Exercise #3

5. September 2022

Exercises marked with a (J) should be handed in as a Jupyter notebook.

Problem 1. (Rounding errors. Only for those enrolled in TMA4130 (4N))

It is known that, for smooth enough functions, the central finite difference approximation of second-order derivatives converges quadratically, that is,

$$\left| \frac{u(x-h) - 2u(x) + u(x+h)}{h^2} - u''(x) \right| = \mathcal{O}(h^2) \, .$$

Computationally, however, this is only attainable for a certain range of h. If we keep reducing h, eventually the convergence will break down, since *machine accuracy* is usually limited to $\mathcal{O}(10^{-16})$ values. Based on that, estimate the mesh size h (in order of magnitude) for which roundoff and approximation errors become comparable.

Problem 2. (Partial derivatives. Only for those enrolled in TMA4135 (4D))

Given the function $f : \mathbb{R}^n \to \mathbb{R}$

$$f(x, y, z) = \sin(xy^2)z^3$$

a) Compute the partial derivatives

$$\frac{\partial f}{\partial x}, \quad \frac{\partial^2 f}{\partial y^2} \quad \text{and} \quad \frac{\partial^3 f}{\partial x \partial y \partial z}$$

b) Compute the gradient $\nabla f(\mathbf{x})$ and the Hessian matrix $Hf(\mathbf{x})$.

c) Compute the directional derivative of f in the direction $\mathbf{v} = (1, 1, 1)/\sqrt{(3)}$ at the point $\mathbf{x} = (\pi, 1, 2)$.

Problem 3. (Fixed-point method)

In this exercise, we will use fixed-point iterations to find the real root of

$$f(x) = 2x^3 - x^2 + 2x - 1 = 0\,. \tag{1}$$

To apply the fixed-point algorithm, we first need to rewrite Eq. (1) as $x = g(x)$, but this representation is not unique. For example, we can write

$$x = \frac{-2x^3 + x^2 + 1}{2} := g_1(x)\,, \quad \text{or} \quad x = \sqrt[3]{\frac{x^2 - 2x + 1}{2}} := g_2(x)\,.$$

In fact, the algorithm's convergence will depend on whether we choose g_1 or g_2 as our $g(x)$. We will experiment with both versions, and you can use the Jupyter notebook 05-*Nonlinear-eqs.ipynb* for help.

a) Find the expressions for $g_1'(x)$ and $g_2'(x)$ and, based on that, answer: would you rather select g_1 or g_2 as your $g(x)$? Why?

b) (J) Set $x^{(0)} = 0.9$ and $g(x) = g_1(x)$, and iterate until $|x^{(k+1)} - x^{(k)}| < 10^{-6}$. What is the solution obtained, and how many iterations are needed to reach that?

c) (J) Now, do the same for $g(x) = g_2(x)$. How many iterations are now needed?

Problem 4. (Fixed-point iterations)

In pipeline design for oil transport, pressure losses must be carefully estimated. They are directly proportional to a positive friction factor f_D, whose inverse square root $x := 1/\sqrt{f_D}$ is given by a non-linear equation. In fact, it is known that x depends only on the Reynolds number R (look up *"Reynolds number"* online, if you are interested in fluid mechanics). For a turbulent flow at a given R, the equation to find x is

$$x + 1.93\ln(1.32x/R) = 0\,,$$

in which $\ln(x)$ denotes the natural logarithm. In this exercise, we will set $R = 5000$ and use fixed-point iterations to approximate x, according to

$$x^{(k+1)} = g(x^{(k)}) \,, \quad \text{with } \; g(x) := -1.93 \ln(1.32x/5000) \,. \tag{2}$$

Let us also define the iteration error as $e_k := |x^{(k)} - x^{(k-1)}|$. Based on that, you are asked to:

a) Compute $g'(x)$ and use the result to determine whether $g(x)$, $x > 0$, is an increasing, decreasing or a non-monotonic function.

b) Calculate the maximum and minimum values of $g(x)$ in the interval $x \in [\mathrm{e}, \mathrm{e}^3]$.

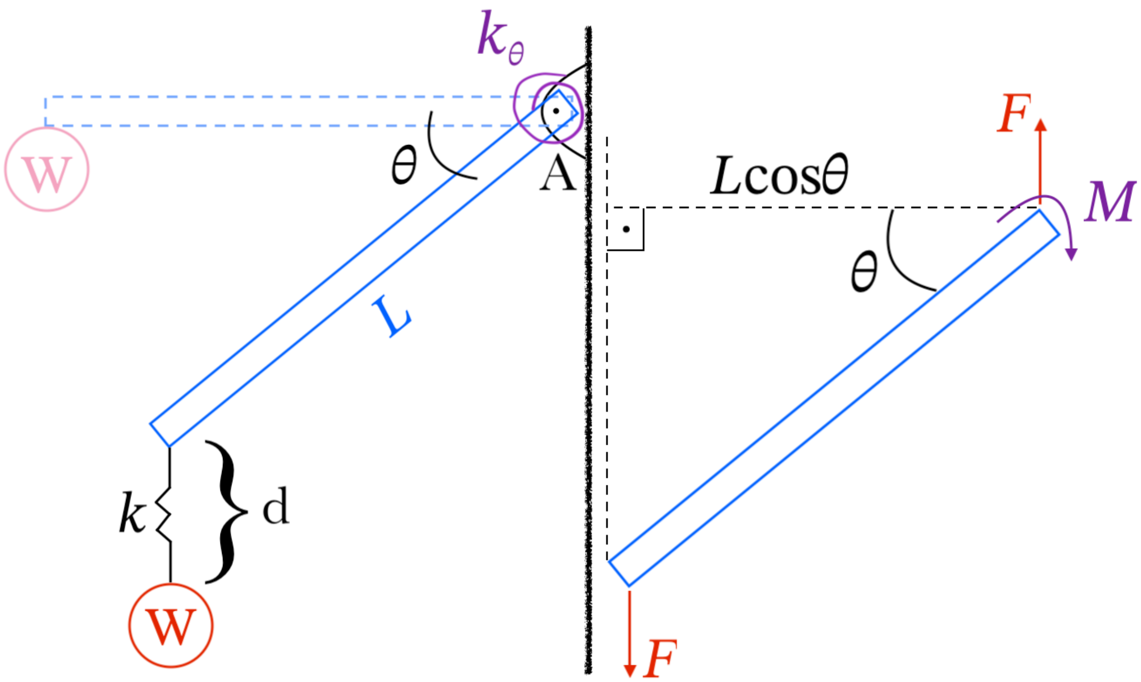
c) Determine the maximum value L of $|g'(x)|$ in the interval $[\mathrm{e}, \mathrm{e}^3]$.

d) For $x^{(0)} = \mathrm{e}^2$, perform the first fixed-point iteration for the solution of Eq. (2).

e) Based on the values of L, $x^{(0)}$ and $x^{(1)}$, find an upper bound for the number of iterations required to reach a tolerance of 10^{-3}.

Problem 5. (Newton's Method)

Figure 1 illustrates the static equilibrium problem of a structural system. At the tip of a weightless bar with length L lies a mass whose weight W pulls the system down. The rigid bar can rotate about its support point A by an angle $0 \le \theta \le \pi/2$. Then, an angular spring responds with an opposing moment $M(\theta) = k_\theta \theta$, with $k_\theta > 0$ being a given constant. The mass and the bar are connected by a linear spring that responds to any separation d with a force $F(d) = kd$, with $k > 0$ being another given constant. Thus, equilibrium of forces for the mass yields $F(d) = W$, while equilibrium of moments for the bar gives $F(d)L\cos\theta = M(\theta)$.



(a) General setup. (b) Free body diagram for the bar.

Figure 1: Static equilibrium problem for a bar-mass system.

To find the *unknown* equilibrium configuration (θ, d), we have to solve a nonlinear system:

$$\begin{aligned} kLd\cos\theta - k_\theta\theta &= 0 \\ kd - W &= 0\,, \end{aligned}$$

in which (W, L, k, k_θ) are all considered as *given* constants. Since the second equation is linear, we can solve it for d and use that result in the first equation. Then, we are left with a nonlinear equation to solve:

$$\cos\theta - \frac{k_\theta}{WL}\theta = 0\,. \tag{3}$$

Since Eq. (3) has no analytical solution, for small θ it is common to use the approximation $\cos\theta \approx 1$. In this exercise, we will compare this *linearised* approach with the nonlinear one. Consider $L = 1$ m, $k = 2$ N/m, $k_\theta = 3$ Nm/rad and $W = 4$ N.

a) Using the simplification $\cos\theta \approx 1$, find θ.

b) Using the θ computed in a) as an initial guess, compute *by hand* the first iteration of Newton's method for Eq. (3).

c) (J) Now, compute a few more iterations until meeting a tolerance of 10^{-6} (you can use the Jupyter notebook 05-*Nonlinear-eqs.ipynb* to make your life easier). Then, compare the linearised θ from a) with the one obtained iteratively: which one is larger?